UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/727,305	11/29/2000	Brad Calder	ENTRPA.013A	7507

20995 7590 06/16/2004

KNOBBE MARTENS OLSON & BEAR LLP
2040 MAIN STREET
FOURTEENTH FLOOR
IRVINE, CA 92614

EXAMINER	
BAUM, RONALD	
ART UNIT	**PAPER NUMBER**
2136	

DATE MAILED: 06/16/2004

8

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/727,305	CALDER ET AL.	
	Examiner	Art Unit	
	Ronald Baum	2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-29 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-29 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 5,7.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1- 29 are pending for examination.

2. Claims 1- 29 are rejected.

Specification

3. The disclosure is objected to because of the following informalities: The attempt to incorporate subject matter into this application by reference to US patent applications only by a title (i.e., page 1, lines 7-9, "METHOD AND PROCESS FOR SECURING AN APPLICATION PROGRAM TO EXECUTE IN A REMOTE ENVIRONMENT", and other locations) is improper because reference to said documents is incomplete without more specific identification (i.e., actual US patent applications numbers).

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 1 recites the limitation "A *system* ..., the *method* comprising:". There is insufficient antecedent basis for this limitation in the claim in that the system is not comprised of "method" elements. The examiner assumes for the sake of applying art that the "method" phrase should be "system".

Claim 12 recites the limitation "The method of claim 7, wherein ...". There is insufficient antecedent basis for this limitation in the claim. The examiner assumes for the sake of applying art that the "claim 7" phrase should be "claim 5".

Claim 22 recites the limitation "The system of claim 19, wherein ...". There is insufficient antecedent basis for this limitation in the claim. The examiner assumes for the sake of applying art that the "claim 19" phrase should be "claim 21".

Claim 24 recites the limitation "The method of claim 19, wherein ...". There is insufficient antecedent basis for this limitation in the claim. The examiner assumes for the sake of applying art that the "claim 19" phrase should be "claim 23".

Claim 27 recites the limitation "The system of claim 25, wherein ...". There is insufficient antecedent basis for this limitation in the claim. The examiner assumes for the sake of applying art that the "claim 25" phrase should be "claim 26".

Claim 29 recites the limitation "The system of claim 25, wherein ...". There is insufficient antecedent basis for this limitation in the claim. The examiner assumes for the sake of applying art that the "claim 25" phrase should be "claim 28".

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-29 are rejected under 35 U.S.C. 102(e) as being anticipated by Kumar, U.S. Patent 6,567,977 B1.

6. As per claim 1; "A system for securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method [system] comprising: a preprocessor module for scanning the application program for code sequences that cause the computer to trap to the operating system and for modifying the code sequences such that the computer does not trap to the operating system [col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39, where the multiple processor system embodiment (col. 2,lines 55-65, col. 4,lines 65-67) whereas any given additional processor is running the shell software; constituting a preprocessor module]; a server computer for receiving at least one application that has been modified by the preprocessor module [the shell trap function]; a network [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33]; and a client computer operably connected to the server computer via the network, wherein the client computer receives the modified application from the server computer, wherein subsequent to receiving the application, the client computer executes the application [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33].".

7. As per claim 2; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: scanning the application for code sequences that cause the computer to trap to the operating system; and modifying the code sequences such that the computer does not trap to the operating system [col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39, where the multiple processor system embodiment (col. 2,lines 55-65, col. 4,lines

65-67) whereas any given additional processor is running the shell software; constituting a preprocessor module].";

Further, as per claim 18; "A system for preventing an application from directly calling an operating system, the system comprising [This claim is the means plus function claim for the method claim 2 above, and is rejected for the same reasons provided for the claim 2 rejection]: means for scanning the application program for code sequences that cause the computer to trap to the operating system; and means for modifying the code sequences such that the computer does not trap to the operating system.".

8. As per claim 3; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: loading the application; marking all of the code pages of the loaded application execute only; and preventing the application from creating executable data during the execution of the application [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation].".

9. As per claim 4; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: preventing the application from creating executable data during the execution of the application [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation]; scanning the

application for code sequences that cause the computer to trap to the operating system; and modifying the code sequences such that the computer does not trap to the operating system [col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39, where the multiple processor system embodiment (col. 2,lines 55-65, col. 4,lines 65-67) whereas any given additional processor is running the shell software; constituting a preprocessor module].".

10. As per claim 5; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: preventing the application from creating executable data during the execution of the application [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation]; and preventing at least one code page of the application from becoming readable and writeable [col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49].".

11. As per claim 6; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: loading the application; marking all of the data pages of the loaded application read and write only [col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49]; and preventing the application from creating executable data during the execution of the application [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation].".

12. As per claim 7; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: preventing the application from creating executable data during the execution of the application [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation]; and preventing the application from modifying executable files or executing any application generated files [col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation].".

13. Claim 8 ***additionally recites*** the limitation that; "The method of Claim 7, additionally comprising: scanning the application for code sequences that cause the computer to trap to the operating system; and modifying the code sequences such that the computer does not trap to the operating system.". The teachings of Kumar suggest such limitations (col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39, where the multiple processor system embodiment (col. 2,lines 55-65, col. 4,lines 65-67) whereas any given additional processor is running the shell software; constituting a preprocessor module).

14. Claim 9 ***additionally recites*** the limitation that; "The method of Claim 7, additionally comprising preventing at least one code page of the application from becoming readable and writeable.". The teachings of Kumar suggest such limitations (col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49).

15. Claim 10 ***additionally recites*** the limitation that; "The method of Claim 7, additionally comprising: loading the application; and marking all of the code pages of the loaded application execute only.". The teachings of Kumar suggest such limitations (col. 2,lines 39-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a marking of executable data memory space).

16. Claim 11 ***additionally recites*** the limitation that; "The method of Claim 7, additionally comprising: loading the application; and marking all of the data pages of the loaded application read and write only.". The teachings of Kumar suggest such limitations (col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49).

17. Claim 12 ***additionally recites*** the limitation that; "The method of Claim 7, wherein preventing the code page of the application from becoming readable and writeable comprises intercepting transparently to the application a request from the application to change the attributes of the code page.". The teachings of Kumar suggest such limitations (col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49, whereas the prevention of the application from modifying the actual computer configuration would constitute a request from the application to change the attributes of the code page.).

18. As per claim 13; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: before the execution of an application program, scanning the application program for code sequences that cause the computer to trap to the operating system; before the execution of the application program, modifying the code sequences

such that the computer does not trap to the operating system [col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39, where the multiple processor system embodiment (col. 2,lines 55-65, col. 4,lines 65-67) whereas any given additional processor is running the shell software; constituting a preprocessor module]; during or subsequent to the execution of the application program, scanning executable data that is created by the application program for sequences that trap to the operating system; and during or subsequent to the execution of the application program, scanning new executable files that are created or modified by the application program [col. 2,lines 39-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a marking of executable data memory space]; and during or subsequent to the execution of the application program, modifying the executable data and the new files such that the application program does not trap to the operating system [col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39].".

19. As per claim 14; "A method of securing an application for execution on a computer [col. 1,lines 13-35,49-64, col. 2,lines 31-46, where an installation program is clearly an application software program], the method comprising: scanning the application for code sequences that cause the computer to trap to the operating system; modifying the code sequences such that the computer does not trap to the operating system; scanning the dynamically generated code that is created by the application for code sequences that cause the computer to trap to the operating system; and modifying the code sequences such that the computer does not trap to the operating system [col. 2,lines 26-col. 4,line 49,65-67, col. 6,lines 4-39, where the multiple processor system embodiment (col. 2,lines 55-65, col. 4,lines 65-67) whereas any given additional processor is running the shell software; constituting a preprocessor module; and further, the real

time operation of the shell software, broadly interpreted, constitutes dynamically generated code].";

Further, as per claim 25; "A system for securing art application for execution on a client computer, the system comprising [This claim is the means plus function claim for the method claim 14 above, and is rejected for the same reasons provided for the claim 14 rejection]: means for scanning the application for code sequences that cause the computer to trap to the operating system; means for modifying the code sequences such that the computer does not trap to the operating system; means for scanning the dynamically generated code, that is created by the application, for code sequences that cause the computer to trap to the operating system; and means for modifying the code sequences such that the computer does not trap to the operating system.".

20. Claim 15 ***additionally recites*** the limitation that; "The method of Claim 14, additionally comprising preventing at least one code page of the application from becoming readable and writeable.". The teachings of Kumar suggest such limitations (col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49,).

21. Claim 16 ***additionally recites*** the limitation that; "The method of Claim 15, wherein preventing the code page of the application from becoming readable and writeable comprises intercepting transparently to the application a request from the application to change the attributes of the code page.". The teachings of Kumar suggest such limitations (col. 1,lines 44-67, col. 2,lines 39-46, col. 4,lines 36-49, whereas the prevention of the application from modifying the actual computer configuration would constitute a request from the application to change the attributes of the code page.).

22. Claim 17 ***additionally recites*** the limitation that; "The method of Claim 14, additionally comprising preventing data pages from becoming executable.". The teachings of Kumar suggest such limitations (col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of preventing the installation software from modifying the actual client computer would constitute a prevention of executable data creation).

23. As per claim 19; "A system for preventing an application from directly calling an operating system, the system comprising [This claim is the means plus function claim for the method claims 3,7 above, and is rejected for the same reasons provided for the claims 3,7 rejection]: means for preventing the application from creating executable data during the execution of the application; and means for preventing the application from modifying executable files or executing any application generated files.".

24. Claim 20 ***additionally recites*** the limitation that; "The system of Claim 19, additionally comprising [This claim is the means plus function claim for the method claim 2 above, and is rejected for the same reasons provided for the claim 2 rejection]: means for scanning the application program for code sequences that cause the computer to trap to the operating system; and means for modifying the code sequences such that the computer does not trap to the operating system.".

25. Claim 21 ***additionally recites*** the limitation that; "The system of Claim 19, additionally comprising means for copying the location of at least one module from a first location to a second location ". The teachings of Kumar suggest such limitations (col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of loading the installation software (i.e.,

module) and the inherent memory management involved as such, broadly interpreted, constitutes module relocation).

26. Claim 22 ***additionally recites*** the limitation that; "The system of Claim 19, wherein the at least one module is a system library.". The teachings of Kumar suggest such limitations (col. 1,lines 13-36, col. 3,lines 18-67, col. 4,lines 36-49).

27. Claim 23 ***additionally recites*** the limitation that; "The system of Claim 19, additionally comprising [This claim is the means plus function claim for the method claim 5 above, and is rejected for the same reasons provided for the claim 5 rejection] means for preventing at least one code page of the application from becoming readable and writeable.".

28. Claim 24 ***additionally recites*** the limitation that; "The method of Claim 19, wherein preventing the code page of the application from becoming readable and writeable comprises [This claim is the means plus function claim for the method claim 12 above, and is rejected for the same reasons provided for the claim 12 rejection] intercepting transparently to the application a request from the application to change the attributes of the code page.".

29. Claim 26 ***additionally recites*** the limitation that; "The system of Claim 25, additionally comprising means for copying the location of at least one module from a first location to a second location.". The teachings of Kumar suggest such limitations (col. 2,lines 47-col. 3,line 32, col. 4, lines 36-49, col. 6,lines 4-33, whereas the process of loading the installation software (i.e., module) and the inherent memory management involved as such, broadly interpreted, constitutes module relocation).

30. Claim 27 ***additionally recites*** the limitation that; "The system of Claim 25, wherein the at least one module is a system library.". The teachings of Kumar suggest such limitations (col. 1,lines 13-36, col. 3,lines 18-67, col. 4,lines 36-49).

31. Claim 28 ***additionally recites*** the limitation that; "The system of Claim 25, additionally comprising [This claim is the means plus function claim for the method claim 5 above, and is rejected for the same reasons provided for the claim 5 rejection] means for preventing at least one code page of the application from becoming readable and writeable.".

32. Claim 29 ***additionally recites*** the limitation that; "The system of Claim 25, wherein preventing the code page of the application from becoming readable and writeable comprises [This claim is the means plus function claim for the method claim 5 above, and is rejected for the same reasons provided for the claim 5 rejection] intercepting transparently to the application a request from the application to change the attributes of the code page.".

Conclusion

33. Any inquiry concerning this communication or earlier communications from examiner should be directed to Ronald Baum, whose telephone number is (703) 305-4276. The examiner can normally be reached Monday through Friday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh, can be reached at (703) 305-9648. The Fax numbers for the organization where this application is assigned are:

After-final (703) 746-7238

Official (703) 746-7239

Non-Official/Draft (703) 746-7246

Ronald Baum

Patent Examiner

AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100